Method for dynamic determination of the delay time of bidirectional databuses

Description of the technical problem

Memory modules, referred to in the following text as DIMM (Dual in-line memory modules), have a defined physical extent. Owing to the finite speed of propagation of electrical signals, the physical extent of the DIMM thus corresponds to a delay time for the electrical signal in order to pass from a source to a sink. This phenomenon is generally referred to as the "line effect", that is to say the "electrical length" of the interconnects is no longer negligible. This is the situation when the highest frequency component which occurs in the signal is at a wavelength which is of the same order of magnitude as the physical extent between the source and the sink.

The higher the data rate on a DIMM, the higher are the frequencies of the frequency components and the shorter are the physical extents for which this line effect must be taken into account. Present memory developments use data rates which lead to major time-critical problems as a result of the subject under discussion. These present memory module developments have the particular characteristic feature of a central integrated circuit (IC) which is mounted on each DIMM. This IC produces the electrical signals for communication with the memory modules locally, that is to say on the DIMM. This basic structure is shown in Figure 1. As can be seen, a number of different signals are indicated there, which are either of different length (DQ/DQS) or else are received simultaneously by a large number of memory modules (CA).

Read access to the memory modules of a DIMM is not the only factor affected by this, but is particularly critical. Read access is distinguished by a command being transmitted via the CA bus (Command and Address

Bus) to the individual memory modules. As can be seen without any difficulty, DRAM4 and DRAM5 are located considerably closer to the data source (HUB) than the modules DRAM0 and DRAM8. It should thus be expected that the read command will reach the DRAM modules 4 and 5 considerably earlier than 0 and 8. The timing diagram in Figure 2 provides an illustration in the form of a graph of this relationship for the DRAMs 4 and 0.

At the time 1, the source (HUB) sends the read command to the DRAM modules. At the time 2, this command reaches the receiver DRAM4. However, since this command is addressed to all the modules, a further delay time is required before the final module (DRAM0) receives the read command at the time 3. After receiving a read command, a dead time passes before the memory modules start to transmit the data. Since all the memory modules are identical, this dead time is also identical for DRAM0 and DRAM4. The dead time at the DRAM4 ends at the time 4, and at the DRAM0 it ends at the time 6. At these times, the DRAM modules start to transmit the required read data. The response from DRAM4 reaches the HUB at the time 5, but the response from the DRAM0 does not reach the receiver module (HUB) until the time 7. Figure 2 shows particularly clearly that a read command which is sent at a specific time 1 leads to a considerable time shift in the responses (times 5 and 7). If the data rate is sufficiently low, that is to say the duration of a single information bit is long in comparison to the time difference 5 and 7, then there is no need to take these effects in account. Owing to the ever wider bandwidth required for memory media, this limit is, however, now considerably exceeded, so that the problem described here needs to be solved.

Previous solution, disadvantages

One normal method for compensating for different delay times is to route the interconnects in a meandering shape on the chip. However, this method is quite unsuitable for this application. Firstly, the meanders require additional space on the DIMM chip, and this is very short. However, a far more serious problem is the fact that the signals do not just have one transmitter and one receiver, but that a number of receivers should be addressed at the same time. This is completely impossible using simple methods since each signal would need to exist two or more times. A signal x which has to be passed from the source to all the DRAM modules would have to exist in versions x0 to x8. Each of these nine signals would then either have no meander at all (for example x0 to DRAM0) or would have a very large number of meanders (for example x4 to DRAM4). If the meandering interconnect routing requires additional space, then the additionally required multiplication of each signal leads to insoluble routing problems. Delay time compensation based on the known meandering routing is therefore impossible on a DIMM.

New solution, advantages

Once the voltage supply for the DIMM modules has been produced, that is to say after the system has been switched on, there is sufficient time to carry out an initialization routine. Since the described problem results from the physical configuration, that is to say the extent, of the arrangement, the effect which needs to be compensated for is a static effect. Furthermore, all the signal sources and sinks are located on the same module, so that there is no need to take into account any external influences. The delay time compensation takes place as an iterative process, which

will now be described in the following text and is illustrated in Figures 3 to 5.

Figure 3:
Once all the dynamic circuit parts of the HUB and of the memory modules have stabilized, for example PLL, DLL etc., the HUB sends a defined command to the DRAM modules. This is done at the time 1. The electrical signal for this command propagates along the DIMM module until it reaches the next receiver, in this case DRAM 4, at the time 2. Since the DIMM is in an initialization routine and is not in the normal operating mode, the dead time (difference between 2 and 3) can be kept very short. Furthermore, there is no need to take any further account of the dead time, since it is identical for all the DRAM modules and only relative delay time differences are relevant. The next DRAM (DRAM4) responds at the time 3 with a unit jump, that is to say it changes the data bus bits at all of its outputs from 0 (low) to 1 (high). This signal transition now once again propagates along the data lines from the DRAM4 until this signal transition is received at the receiver at the time 4. At the time 5, the initialization command from the time 1 also reaches the DRAM which is furthest away (in this case DRAM0). At the time 6, this then also changes its data bus bits from 0 (low) to 1 (high). At the time 7, the HUB receives this signal change in the data bits of the transmitter (DRAM0) which is furthest away.

So far, no significant information has yet been obtained about the delay time of the individual data bits. However, this is achieved if, at the time 1, not only is the command sent but also at the same time a

type of "stopwatch" is started in all the receiving data lines of the HUB.

This stopwatch is represented by a controllable integrator. Figure 4 shows the essential details of the integrator. One important feature of the integrator is a reference value (U_{REF}). As soon as the integrator has exceeded this value, an indication is produced, that is to say an output signal changes its state. However, the most important feature of the integrator is that the gradients can be controlled by a binary word. The integrator is started at the time A, and it exceeds the reference value at the time B. The time difference between A and B depends on the gradient of the integrator. The shallower the gradient, the greater is the time period before the reference value is exceeded. This is illustrated by the times B_1, B_2 and B_3.

Principle of Operation

In order to understand the principle of operation, a brief description should first of all be given of what the initialization routine is intended to achieve. A time variable must be determined for each data line between the HUB and the connected memory module, in order to compensate for the different delay times for further processing. For this purpose, each data line of the HUB has its own controllable integrator as described above, and as is illustrated in Figure 4. As soon as the command is sent to the memory module, that is to say at the time 1 in Figure 3, each data line starts its own integrator. As soon as the associated data bit changes from 0 to 1, the integrator is stopped. If the reference value U_{REF} had already been exceeded at this stopping time, the line was slower than assumed and the measurement must be repeated.

However, this is now done with a shallower integrator gradient. With one integrator gradient, the data signal is now received at an earlier time than the integrator requires to exceed the reference value. Since the gradient of the integrator is controlled by a binary word, this binary word at the same time represents a measure of the delay on the data line. This process is now repeated until all the data lines have been measured and a specific binary delay time word has been determined for all of the data lines. This value is now used to additionally delay all the data lines such that the data within the HUB is subject to a standard delay, and the time consistency is ensured once again.

Figure 5 once again shows a simple outline overview of the method of operation. The source in the module on the left sends a command to the command and address bus to carry out the delay time measurement. However, this is indicated only by a sudden change. At the same time, this event indicates the start condition for all the controllable integrators in the data line circuits. The different delay times to the individual sinks are represented by the line elements and are illustrated with delay times t_{12}, t_{23} etc. The sinks cause the sources in the addressed modules to send a measurement pulse. This is once again perceived by the data lines and indicates the stop condition for the integration, and initiates a check as to whether the integrator has already exceeded the reference value. All the delay times can thus be determined by iteration.

Figure 6 shows the time sequence for the delay time measurement. Once the supply voltage has been applied (power up), all the modules start to carry out their self tests. If these have been successful, the modules

enter the initialization routine for the delay time measurement which is the subject matter of this invention report. Once the delay time measurement has been carried out and the associated compensation values have been determined, the memory module can change to the normal operating mode.

Advantages of the Algorithm

- Simple implementation
- Feasibility both with analog and mixed signal methods as well as with digital circuit concepts
- Small area and low power requirement
- No need for high-frequency clock signals for counting algorithms
- Capability for single-ended compensation (a determined value of the de-skew can be used in inverted forms as a pre-skew, which considerably simplifies the circuit complexity of the DRAM modules)

Essence of the invention, principle

It is of major importance that the method described here is a flexible method. At the time when the circuit parts involved are produced only the orders of magnitude of the delay time to be compensated for are required. There is no need for detailed analysis of the physical design. The method is sufficiently flexible to be adapted to the conditions after assembly. It is also a fast and simple method which can be carried out during the switch-on phase (boot time) without this resulting in any need to accept regular reductions in performance.

Since each line has its own controllable integrator, the concept can be extended to any desired number of

data lines. Furthermore, this allows parallel processing, that is to say all the data lines are processed at the same time. There is no need to process one data line after the other. In the case of broad data lines (for example 72 bits on one DIMM), this is the major reason why the method can be carried out so quickly.

The control logic for the individual integrators, whose main object is to check whether the last delay time measurement was successful, can be implemented centrally. This means that the complexity for these circuit parts is required only once. However, with present semiconductor technologies, this represents only a minor advantage. On the other hand, each data line can also include its own control logic so that it can act completely individually. This may be of interest for data transmissions in which the data bus width is intended to be enlarged dynamically in order successively to increase the total data throughput, and to match it to the requirements.

The integrator is started at the same time that the delay time command is transmitted, so that there is no need for complex detection methods to determine the start time.

The value which represents the delay time need not be produced by high-frequency counting pulses but is available simply from the gradient of the integrator. The delay time measurement merely determines whether the previously assumed value, that is to say the instantaneous gradient, is or is not correct. A step-by-step iteration process is used to approach the

value at which the delay time measurement is successful.

Verification of the invention in competitor products

Verification can be obtained only by detailed knowledge of the competitor product, that is to say by reengineering. If there is a suspicion of patent infringement, this can be done relatively easily by measurement of the module. A competitor product would thus likewise start by transmitting measurement commands after application of a supply voltage, and would continue this until measurement signals arrive on the data lines at suitable time intervals. The principle of this invention report would thus be infringed. Whether a specific implementation would also be infringed can be determined only by reengineering.